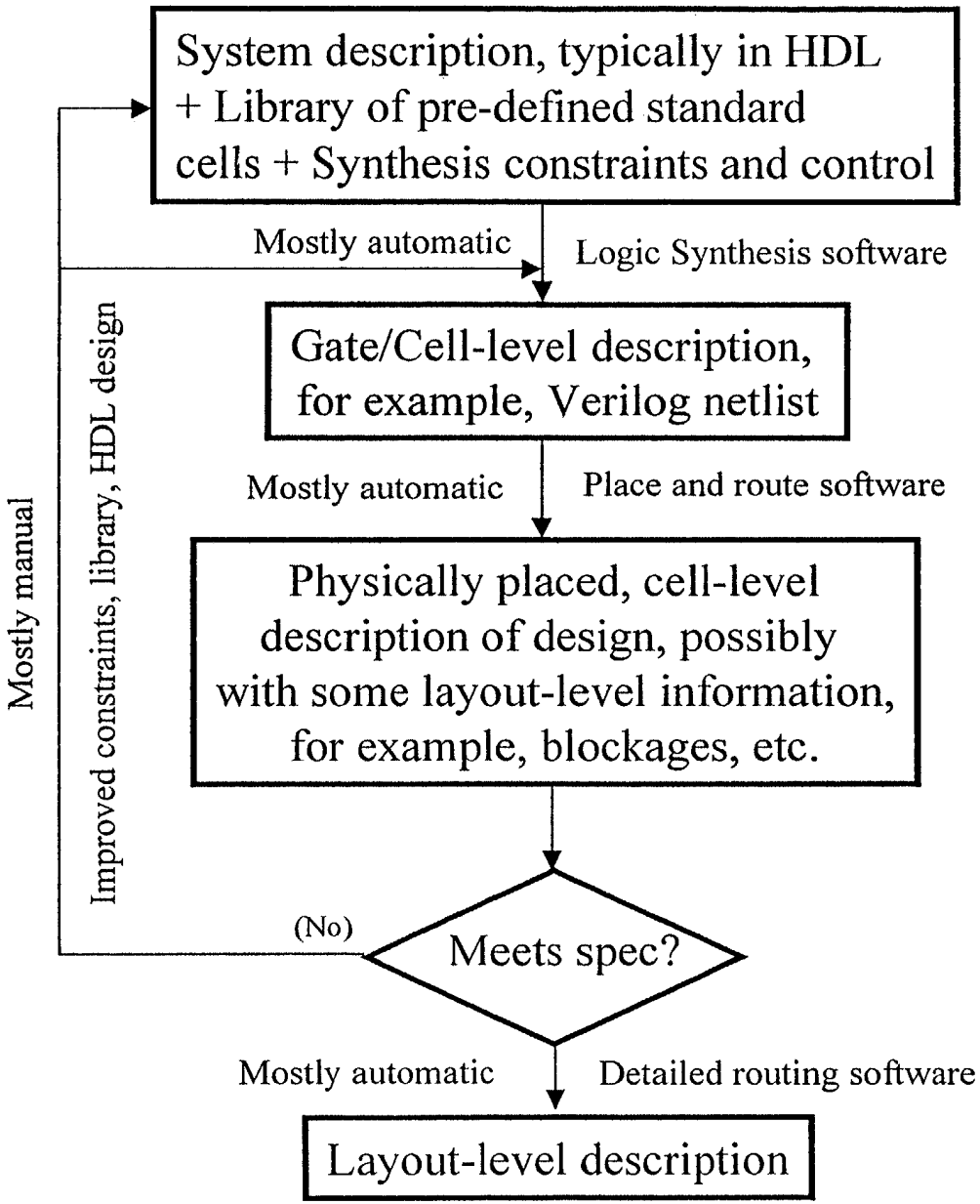

System description, typically in HDL + Library of pre-defined standard cells + Synthesis constraints and control
Mostly automatic
Logic Synthesis software
Gate/Cell-level description, for example, Verilog netlist
Mostly automatic
Place and route software
Physically placed, cell-level description of design, possibly with some layout-level information, for example, blockages, etc.
Meets spec?
(No)
Mostly manual
Improved constraints, library, HDL design
Mostly automatic
Detailed routing software
Layout-level description

Fig. 1

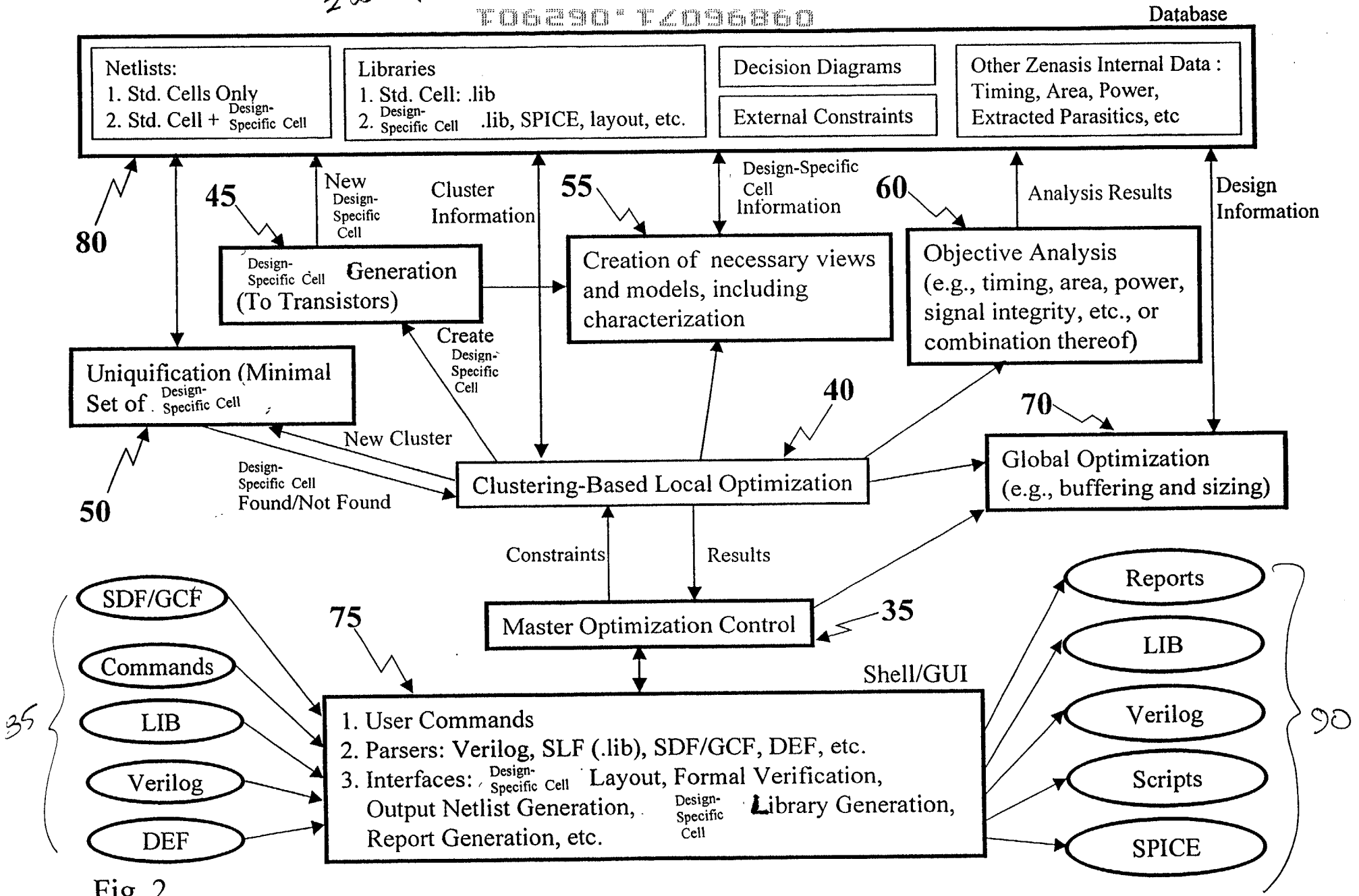


Fig. 2

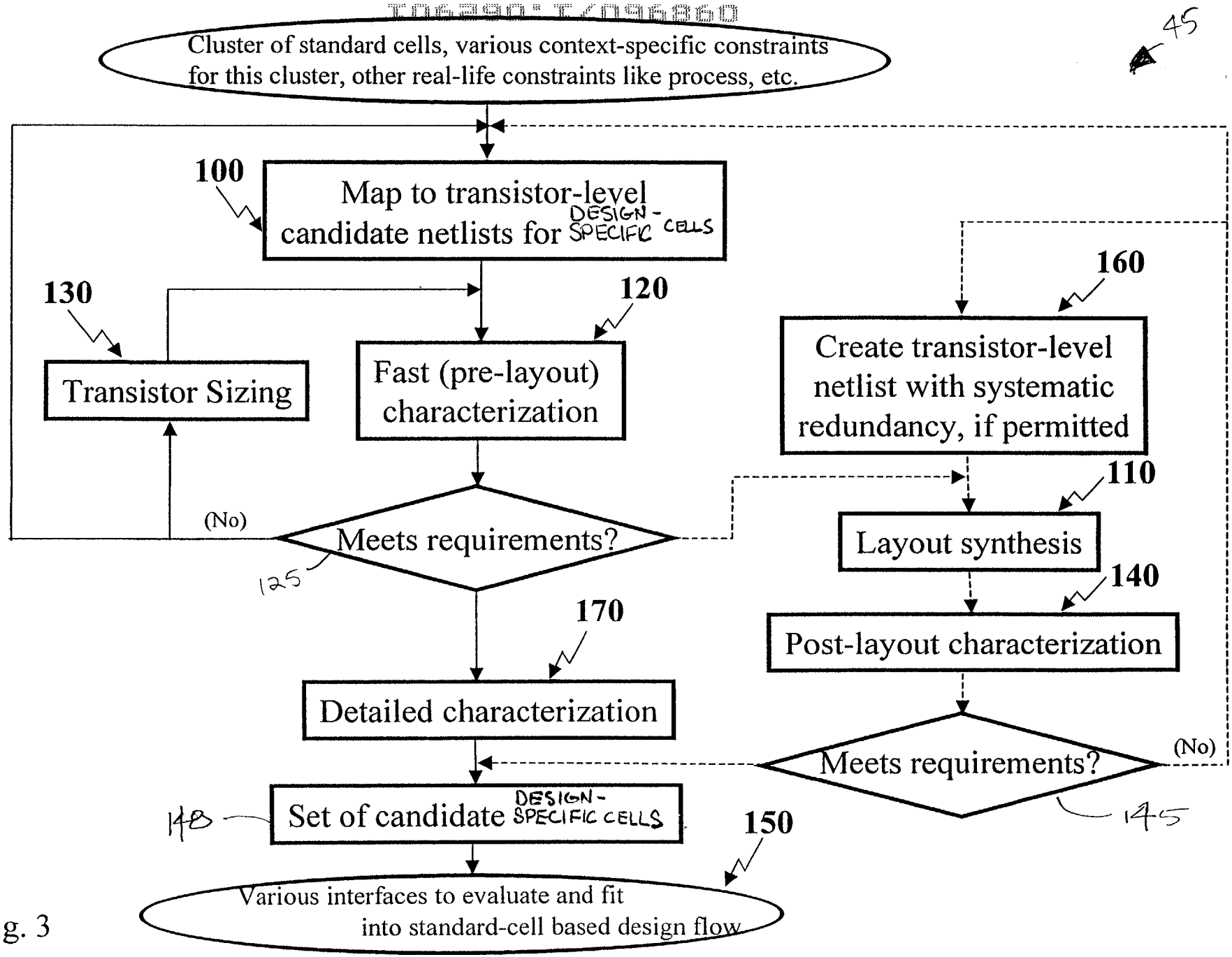

Cluster of standard cells, various context-specific constraints for this cluster, other real-life constraints like process, etc.
45
100
Map to transistor-level candidate netlists for DESIGN-SPECIFIC CELLS
130
Transistor Sizing
120
Fast (pre-layout) characterization
160
Create transistor-level netlist with systematic redundancy, if permitted
110
Layout synthesis
140
Post-layout characterization
(No)
Meets requirements?
125
170
Detailed characterization
(No)
Meets requirements?
145
148
Set of candidate DESIGN-SPECIFIC CELLS
150
Various interfaces to evaluate and fit into standard-cell based design flow

Fig. 3

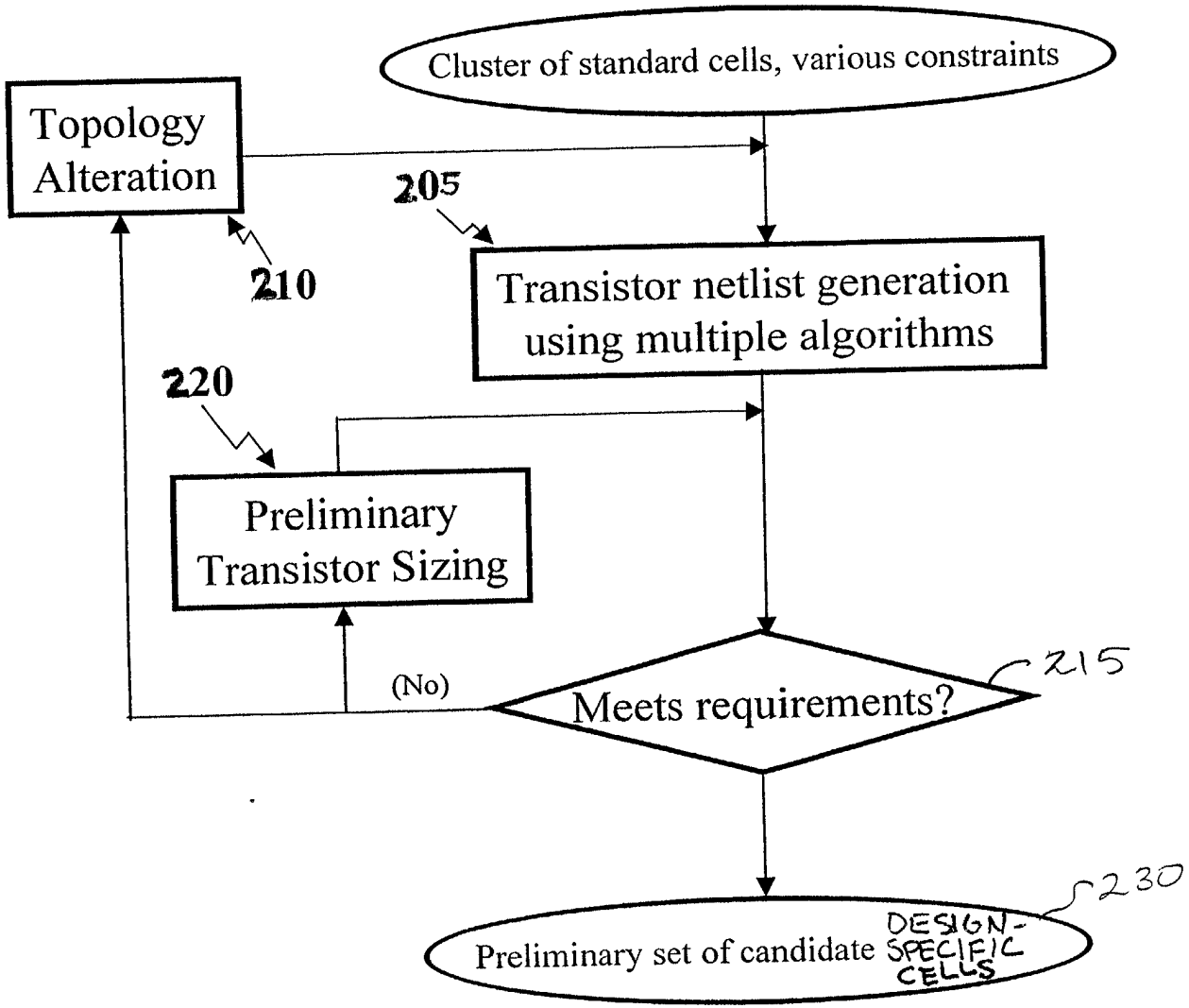
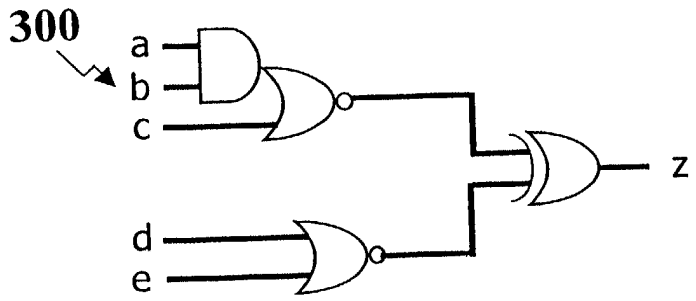


Fig. 4



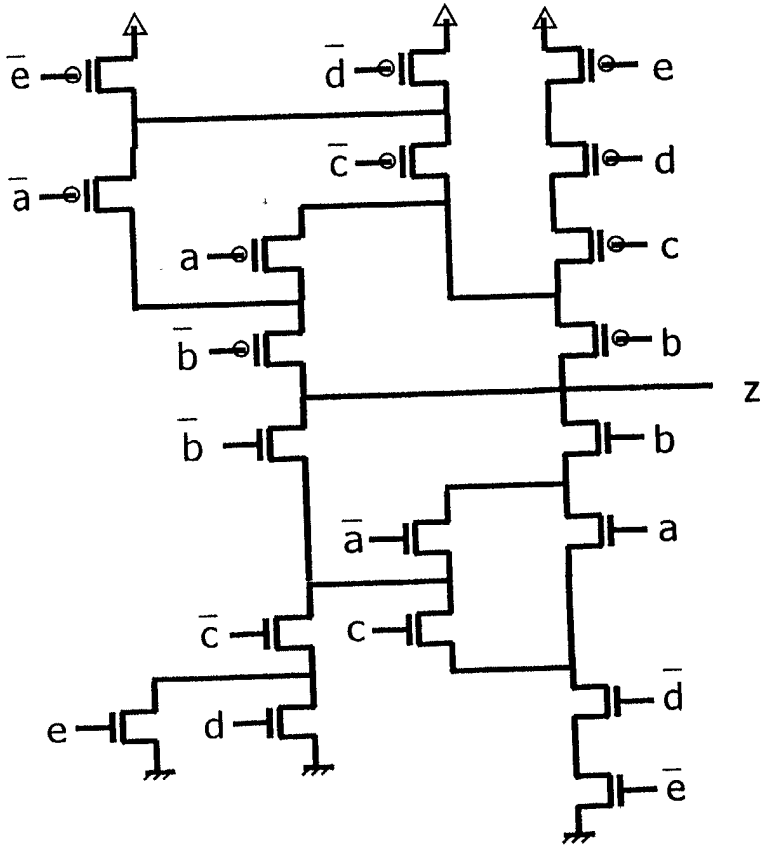
5a

Critical path to z through b

⇒ **Delay from b to z : 0.33 ns**

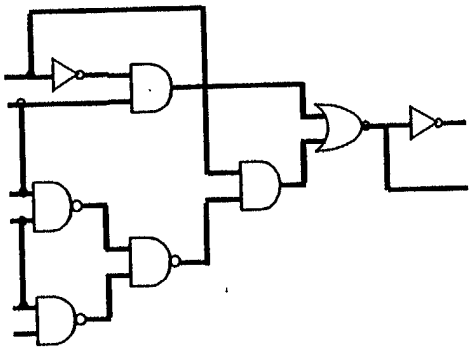
⇒ DESIGN-SPECIFIC CELL **delay : 0.14 ns**

5c

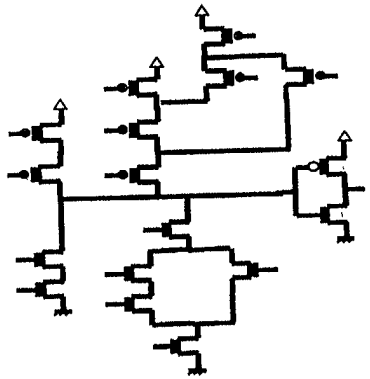


5b

Fig. 5



6a

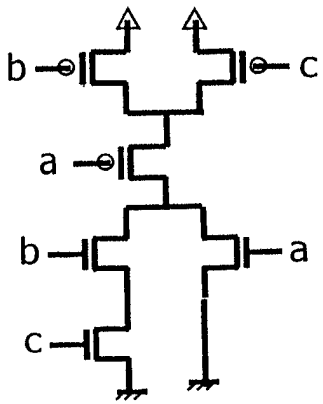


6b

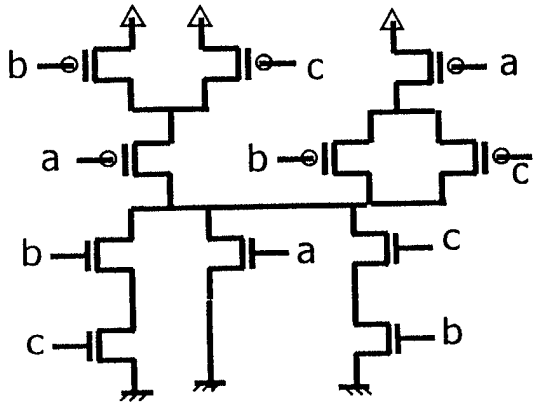
Criteria	Original	Optimized
#of Cells	8	1
#of Transistors	32	17
#of Wires (incl. I/O)	12	5

6c

Fig. 6



Typical standard cell implementation with no systematic redundancy -- input a usually has fastest propagation through cell, c slowest



Implementation of same functionality with systematic redundancy -- inputs a and c usually have comparable propagation delay through module

Fig. 7